John T. Anderson

Engineering Note

Date: January 8, 2002
Rev Date: January 8, 2002

Project: Analog Front End board
Doc. No: A1020108

Subject: Revisions to 1553 interface after yet more meetings….

Introduction

Despite numerous "final solutions" having been promulgated before, continued resistance to handling any form of BUSY response from the AFE to the 1553 system has resulted in a new version of the 1553 interface logic. Paul Rubinov has designed some new logic which is documented here. The intent and desire is to create an interface where no BUSY response ever occurs.

General Algorithm

On January 3rd, 2002, Paul sent out a note describing his viewpoint of the system, copied here verbatim.

Proposal:

1. Modify the 1553 controller CPLD and the Helper CPLD such that they time multiplex so that BOTH are guaranteed a valid transaction, no matter when the transactions happen.
2. Add 2 16 bit command counters to the PIC: one counter is incremented when the PIC receives a command, the other is incremented when the PIC completes a command.

The system would function as follows:

The PIC is polling the command execute flag, except when it stops polling in order to execute a temperature control update, which happens about once a second and takes about half a second (so the PIC is not polling 50%) of the time.

1. The user reads the command queue to make sure it is empty.
2. If it is empty, the user writes the command parameters and the command.
3. The user reads the command counters and stores their values for future reference. If the two values are equal, proceed to the next step, otherwise wait.
4. The user writes 0xFF to the command execute flag location.
5. (optional) the user reads the values of the command counters. The command processing is completed when both counters have incremented by one.

THIS ASSUMES THAT ADMINISTRATIVE CONTROLS ARE IN PLACE WHICH PREVENT MULLTIPLE USERS FROM TRYING TO EXECUTE COMMANDS ON THE SAME AFE AT THE SAME TIME.

Note that there is no problem with the temperature control and monitoring area because direct access by the 1553 of the locations does not prevent the PIC from always succeeding in reading and/or writing valid data in the same locations.

Specific Details of Actual Implementation

In an effort to provide maximum functionality, the ALT_BUSY version of the 1553 interface written by J.T. Anderson in December 2001 was used as the starting point of the design. This was further modified per Paul's algorithm to yield a device with the following features:

1. Three of the GREEN_WIRE bus bits are used as new controls to the 1553 PLD:

   1.1 GREEN_WIRE[5] is the PIC BUSY, active HIGH, which is driven to the 1553 PLD from the Helper PLD. The microcontroller may set and reset this bit as desired.

   1.2 GREEN_WIRE[6] is the READ_SLOW_CLOCK input (active LOW), driven from the Helper PLD to the 1553 PLD. This signal, asserted by the microcontroller, causes the 1553 PLD to drive the current value of the slow (realtime) clock onto the PIC data bus, to aid in synchronization of the temperature control loop.

   1.3 GREEN_WIRE[7] is the RESET_SLOW_CLOCK input (active LOW), driven from the Helper PLD to the 1553 PLD. This signal, asserted by the microcontroller, causes the slow (realtime) clock to be reset to zero.

2. The RESET pin of the 1553 PLD, which is connected to the board-wide POWER_ON_RESET net, is redefined as a BIDRECTIONAL signal, not just an input. In response to a very particular 1553 transaction, the 1553 PLD will *drive* this signal low long enough to cause the *entire AFE to experience the exact same reset pulse as it would 'see' at initial power-up.*

   2.1 This RESET-on-demand *is not,* repeat, *is not exactly the same as cycling the power of the board.* The following WILL OCCUR:

      2.1.1. The microprocessor will re-start and perform the initial initialization of some areas of the dual-port RAM.

      2.1.2. Power to all MCMs will be turned OFF.

      2.1.3. All PLDs will reset to their power-on state.

   2.2 The following things WILL BE DIFFERENT:

      2.2.1. Although the VREF and SIFT threshold *voltages* will drop to power-on states (due to the *reference voltages* being turned off when the MCMs are turned off), these voltages *will revert to their last programmed state upon turning the MCMs back on.* The reset signal generated by the 1553 interface *does not reset the DAC demand values to zero.*

      2.2.2. The VLPC Bias Voltage DACs have a separate reference voltage. As such, *the VLPC Bias voltage WILL NOT BE AFFECTED BY THE 1553 RESET.*

      2.2.3. The Cryostat Heater DACs have a separate reference voltage. As such, *the Cryostat Heater DACs WILL CONTINUE TO DRIVE HEATER POWER.*

      2.2.4. The re-start of the Microprocessor, however, will set the Heater Demand values to all zeroes and will *disable* temperature control and monitoring.

   2.3 A specifically obscure 1553 transaction, well different from any normal AFE transaction, was chosen as the 'reset request'. In order to ask for a reset, a *write* transaction to subaddress 21 (decimal) must be used, with a word count of *exactly* nine words. The data contained in the nine words is immaterial. Any other access to subaddress 21 will be ignored.

      2.3.1. Since the board-wide reset also resets the 1553 interface PLD, obviously, no proper status word response to the 1553 controller should be expected.

      2.3.2. To determine if the reset actually occurs in response to the reset request, software should load some known data in one of the areas of memory cleared by the microcontroller upon power-up reset and verify that the data has in fact been cleared. Addresses 0x0050 – 0x005F are cleared upon power up; of these, addresses 0x0058 through 0x005D are currently unused by the PIC firmware and may be used for this test.

3. GREEN_WIRE[5] (the PIC BUSY) propagates back to the 1553 controller, not via the status response of the AFE, but only by data value changes when reading the register located at subaddress 18. A BUSY does NOT block the 1553 from accessing the dual-ported RAM. Previous proposals would have blocked DPRAM access during the BUSY times but this scenario is addressed by Paul's new logic, described later in this document.

   3.1 The data value received when reading subaddress 18 (decimal) will change dependent upon whether the PIC is busy or not. If the PIC is BUSY, the value read will always be 0x803C. If the PIC is not busy, the data word will be 0xnn5A, where 'nn' is the current value of the slow (realtime) clock.

   3.2 The BUSY status response in the Status word as defined in the 1553 specification will never be asserted.

4. Nothing blocks 1553's ability to write to the DPRAM. The WE* will occur as a 1 usec pulse regardless of what the PIC is doing. The assertion of the *address* to the DPRAM, however, is qualified such that it only occurs for the 3 usec window during which the write will occur; otherwise the address is forced to 0x0000, a location never used by the PIC.

5. Similarly, nothing blocks 1553's ability to read the DPRAM. The 1553 interface as coded requires that the data from the DPRAM be stable for 16 usec while it is serialized to the cable.

   5.1 The PIC's write cycle to the DPRAM is extended from the ~400 nsec cycle of the earlier revisions to a 20 usec write cycle to insure that there is a guaranteed 1 usec somewhere within the 20 usec where the WE* is asserted and there is no address conflict. *This requires a different version of the PIC firmware than is presently implemented in boards on the platform.*

   5.2 The architecture of the DPRAM guarantees that if the PIC's write cycle begins after the read by 1553 has begun, the output buffer of the DPRAM will hold the 1553's data constant until the address collision is removed.

   5.3 Thus, the PIC write always succeeds and the 1553 data is never garbled. However, if the PIC write occurs at the same time as the 1553 read, the 1553 reads "stale" data.

6. The PIC's access to the DPRAM is time-sliced by modulating the CE* inputs of the DPRAM at 6MHz (period = 167 nsec).

   6.1 PIC writes persist for 20 usec, as given above. The guaranteed 1 usec of PIC access allows multiple redundant writes to occur from the CE* timeslicing.

   6.2 PIC reads of DPRAM data give one to 1.5 instructions of setup time after the XFR_STRB is asserted before the data is actually transferred. One instruction at 12 MHz is approximately 300 nsec. This insures that at least two or three reads of the RAM have occurred prior to the PIC looking at the data bus. The natural capacitance of the bus does not allow it to recover from any zeroes driven by the RAM during the 167 nsec timeslices, so in effect a 'latch' is created that holds persistent data for the PIC to read.

   6.3 The 167 nsec timeslices caused by the 6MHz clock also allow at least a couple write cycles (minimum write time = 25 nsec) from 1553 to occur, so a read by the PIC cannot block a write by 1553. Similar to the scenario in (5) above, the PIC might get the new or the old data; however, reading old data instead of new will only result in a delay in processing time. The PIC might not 'see' the command on this cycle if a conflict occurred, but it will 'see' the data on the next cycle.

Specific Details of PIC Firmware Modifications

In addition to the timing and hardware interface issues listed previously the changes to the PIC firmware also introduce new DPRAM locations which may be used for diagnostic purposes.

1. A new 16 bit counter of 'command triggers received' is now implemented, which counts the number of times the PIC has responded to the condition of DPRAM location 0x0057 (as viewed from 1553) being non-zero. This counter is written to DPRAM at location 0x0249.

2. A new 16 bit counter of 'command queues processed' is now implemented, which counts the number of times the PIC has completed processing a command queue. This counter is written to DPRAM at location 0x024A.

3. It is expected that external processes will utilize these counters to determine if other concurrent processes are sending commands to the same AFE.

   3.1 Most of the time the two counters should read the same value. If the 'triggers received' counter is one greater than the 'queues processed' counter, this should be interpreted as the PIC is currently processing the last received queue of commands.

   3.2 The two counters should both increment by one (first the one counter, then somwhat later the other) every time a process sends a queue of commands to the AFE. If the counters have incremented by more than one, some other process must have issued commands to the same board.

   3.3 A process should expect that the counters will both increment quite quickly after the command queue has been sent. Failure to see either counter increment would presumably indicate that the command queue was either never sent or never processed. Examination of other DPRAM locations should be able to determine which has occurred.

      3.3.1. Reading DPRAM location 0x005F (the heartbeat) a couple times will indicate if the PIC is somehow locked up and unable to change the heartbeat.

      3.3.2. Reading DPRAM location 0x0060 (the queue process flag) will indicate the command processing state of the PIC:

         3.3.2.1. Reading the value 0xFFFF indicates that the PIC is not executing commands.

         3.3.2.2. Reading the value 0xA5A5 indicates that the PIC is currently processing a command queue.

   3.4 The PIC BUSY status may still be read by looking at subaddress 18; if the PIC is stuck at BUSY all the time (that is, continuously for more than a few seconds), this may also indicate a firmware error.

Implementation into AFE Boards

While it is not strictly necessary to swap in PICs with new firmware at the same time the PLDs are updated – the design of the PLDs is backwards compatible – the collision avoidance mechanism requires that both be changed in order to obtain full functionality. If boards in the platform receive the PLD upgrade but the PICs are left untouched, collisions can still occur.